Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 52 (2015) 244 – 251

Procedia Computer Science

6th International Conference on Ambient Systems, Networks and Technologies, ANT 2015

Dendritic Cell Algorithm for Mobile Phone Spam Filtering

Ali A. Al-Hasan[a], El-Sayed M. El-Alfy[b,*]

[a]*Saudi Aramco, Dhahran, Saudi Arabia*

[b]*College of Computer Sciences and Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia*

Abstract

With the revolution of mobile devices and their applications, significant improvements have been witnessed over years to support new features in addition to normal phone communication including web browsing, social networking and entertainment, mobile payment, medical and personal records, e-learning, and rich connectivity to multiple networks. As mobile devices continue to evolve, the volume of hacking activities targeting them also increases drastically. Receiving short message spam is one of the common vectors for security breaches. Besides wasting resources and being annoying to end-users, it can be used for phishing attacks and as a vehicle for other malware types such as worms, backdoors, and key loggers. The next generation of mobile technologies has more emphasis on security-related issues to protect confidentiality, integrity and availability. This paper explores a number of content-based feature sets to enhance the mobile phone text messaging services in filtering unwanted messages (a.k.a. spam). Moreover, it develops a more effective spam filtering model using a combination of most relevant features and by fusing decisions of two machine learning algorithms with the Dendritic Cell Algorithm (DCA). The performance has been evaluated empirically on two SMS spam datasets. The results showed that significant improvements can be achieved in the overall accuracy, recall and precision of spam and legitimate messages due to the application of the proposed DCA-based model.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Mobile Technology; Smartphones; Short Message Service (SMS); Dendritic Cell Algorithm (DCA); Spam Detection and Filtering; Application Security.

1. Introduction

Nowadays, with the advances in mobile technology, end users are accessing their emails, surfing the world-wide web, making video & voice calls, using text chatting, gaming and more through their smartphones. The number of mobile users is increasing significantly over time with almost seven billion cellular subscriptions worldwide[1]. Mobile devices are now likely to contain personal and confidential information such as credit card numbers, contact lists, emails, medical records and other sensitive documents. Unlike desktop applications, effective security controls to protect mobile devices are not mature enough and is an active area of research. This can be attributed limited resources and processing power, and lack of knowledge and awareness of many end users regarding protection mechanisms. These reasons and more are making mobiles very attractive to cyber attacks. Hackers can utilize the compromised

* Corresponding author. (On leave from Tanta University, College of Engineering, Egypt).
E-mail address: alfy@kfupm.edu,sa

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2015.05.067

mobiles to make calls to premium numbers without the end-users' permission, stealing contact data, or participating in botnet activities.

Exchanging short text messages (SMS) among mobile phones is very convenient and frequently used for communication on a daily basis. Subsequently, the number of unwanted SMS messages (spam) is growing. In 2012, there were 350,000 variants of SMS spam globally[2]. SMS has been considered a serious security threat since early 2000s[3]. For example, hackers can send phishing attacks to collect confidential information or launch other types of attacks. The risk of SMS spam could be operational or financial loss. It is getting easier to target end users through SMS than electronic mails (emails) since the mail service is more mature, and more effective email spam fighters have been developed and deployed by service providers and users. Unfortunately, this is not the case with the SMS spam. The controls that are used by mobile phones to block SMS spam are not as effective as email anti-spammers. It is a challenging task since SMS messages have limited sizes which means less statistically-distinguishing information. Recently, several methods have been investigated to detect SMS spam, including content-based approaches[3–7]. However, the accuracy is still relatively low and further research is required to investigate new features and new ways of calculating and utilizing them.

In this paper, we analyze several feature sets and study their impact on two machine learning algorithms. Then, we combine the top two relevant feature sets and build a more effective model. Inspired by the danger theory and the immune-based systems, we propose a novel approach based on the Dendritic Cell Algorithm (DCA) for fusing the results of Naïve Bayes (NB) and Support Vector Machines (SVM). DCA is a relatively recent approach in machine learning[8]. Using two SMS datasets, we evaluate and compare the effectiveness of the individual feature sets and the proposed fused model.

The remainder of this paper is organized as follows. Section 2 describes the methodology and Section 3 presents the empirical analysis and results. Finally, Section 4 concludes the paper.

2. Methodology

The generic framework for fighting against textual SMS spam is typically treated as a document categorization problem where individual messages are preprocessed and represented by feature vectors. Then, statistical or machine learning models are built using a training corpus to determine the category for each received message to be spam or legitimate (ham). Differences among various approaches are mainly in how messages are transferred to feature vectors and how classification takes place. The details of the main phases of the proposed model are provided in the following subsections.

2.1. Corpus Analysis and Representation

2.1.1. Enrichment

To enrich the SMS, we added two types of semantic information tagging: part-of-speech (POS) and recognized entities tags. The POS tags are the linguistic categories of words. We assign the POS tags using the Penn Treebank tag set (`http://www.cis.upenn.edu/~treebank/`). Examples of the possible tags are nouns, verbs, adjectives and adverbs. We only extracted the part-of-speech tags for the first and last terms in each message as features since they describe embedded grammatical structure that is unlikely to vary for each spammer or author[9]. The other type of tags corresponds to recognized named entities using the OpenNLP model (`https://opennlp.apache.org/`). These entities include location, organization, money, date, person and time[10].

2.1.2. Preprocessing

The preprocessing phase includes the following steps. First, the SMS message is converted into lowercase characters before being passed to the next stage. Second, each SMS message is treated as a string and then divided into distinct tokens (words). Third, each word is reduced to its root by removing all suffixes and prefixes such as 'tion', 'ing' and 'er'. We used the Porter stemming algorithm to achieve this task[11].

2.1.3. Feature Extraction

Feature extraction is a very crucial task for the SMS classification. It should not require complex analysis in order not to significantly delay the messaging service. But extracted features should also be highly correlated to the message

category to enhance the spam detection accuracy. As a result, each message is represented with a vector denoted as $X = (x_1, x_2, x_3, \ldots, x_m)$, where m is the number of features and x_i for $i = 1, ..., m$ represents the weight of the i-th feature to that message. In our work, we extracted and evaluated the following feature sets for SMS spam detection:

- *URL Link*: We normalize all URL links within SMS messages by replacing them with a single word (e.g. httpLink). We consider the number of URLs in the SMS message as a feature since malicious spam SMS likely asks the user to click on a link to visit a website for a prize or to download an application.
- *Spam Words*: A set of words and phrases are most commonly used by spammers[6]; see Table 1 for examples. We used the number of spam words that exist in an SMS as a feature . Our list consists of 350 terms collected from various sources and blogs that are publicly available on the web.
- *Emotion Symbols*: The existence of emotion symbols and icons may be a good indicator for legitimate SMS messages. Examples of these symbols are happy, angry or sad faces. We used regular expressions to extract these symbols.
- *Special Characters*: Spammers might use special characters for various reasons such as by-passing simple filters based on keywords. For example, the dollar signs "$$$" can be used instead of money in prize or finance related messages. We used regular expressions to extract these features.
- *Message Metadata*: This feature set includes message length, which is the overall byte length of SMS, number of tokens and average token length.
- *Function Words or Grammatical Words*: These are non-content words that have little lexical meaning or have ambiguous meaning, but exist to explain structural or grammatical relationships with other words within a sentence or specify the mood or attitude of the author. Function words form a closed class of words that is fixed and has a relatively small size. For example, Koppel and Ordan[12] used 300 function words for the English language from LIWC[13]. Function words are lexically unproductive and are generally invariable in form. Examples of function words are prepositions, pronouns, determiners, conjunctions, auxiliary verbs, and particles; see Table 2. We evaluated function words features because they are very unlikely to be subject to conscious control by an author. This is due to their high frequency of use and highly grammatical role[14]. We relied on the word list available in[15].

In addition to these feature sets, we included two other feature sets calculated during the enrichment phase which are POS tags of the first and last terms in each SMS and the named entity tags (referred to as All tags).

Table 1. Examples of common spam words and phrases.

credit, loan, bills, info, money, investment, discount , win, order now, sign up, clearance, earn, free gift, free samples
dating, find, guess, statement, private, dear, partner, singles, fast cash, incredible deal, free info, satisfaction, buy direct
call free, call now, camcorder, phone, cards, extra inches, cialis, viagra, spa, beauty, money back, click here, act now
prize, guaranteed, claim, cash, no fees, limited time, life insurance, mortgage, amazing, 100% satisfied, 100% free

Table 2. Examples of function words.

Class	Size	Examples
Prepositions	124	of, at, in, on, for, without, between, besides, close to, down
Pronouns	70	he, she, you, him, her, our, anybody, it, one
Determiners	28	a, the, all, both, either, neither, some, those, every
Conjunctions	44	and, after, hence, however, that, when, while, although, or, yet
Auxiliary and modal verbs	17	may, had better, used to, might, shall, be able to, can, must
Quantifiers	>86	no, none, one, two, much, many, the whole, part, various

Algorithm 1: Generation of DCA Signals

Data: SVM_c and NB_c decisions; SVM_{cf} and NB_{cf} confidences
Result: signals: $PAMP$, $Safe$, $Danger$
begin
  $PAMP$=0, $Safe$=0, $Danger$=0;
  if SVM_c==NB_c **then**
    if SVM_c=="*Spam*" **then**
      $PAMP$= Max(SVM_{cf},NB_{cf});
    else
      $Safe$= Max(SVM_{cf},NB_{cf});
    end
  else
    $Danger$= Avg(SVM_{cf},NB_{cf});
  end
end

Algorithm 2: DCA Learning Algorithm

Data: Antigens and Signals($PAMP$, $Safe$, $Danger$)
Result: Antigens and their $MCAV$ values
begin
  initialize DC;
  while *there is input* **do**
    if *Antigen* **then**
      Expose DC to Antigen;
    else if *Signals* **then**
      calculate K and CSM;
      update DC;
    end
    if *DC lifespan* < 0 **then**
      reset DC;
    end
  end
  for *each Antigen type* **do**
    calculate $MCAV$
  end
end

2.2. DCA-Based Classification

The Dendritic Cell Algorithm (DCA) is a recent immune-inspired classification algorithm developed based on the behavior and function of Dendritic Cells (DCs) in the biological immune system[8,16]. The algorithm was successfully applied to solve a number of classification problems in various domains, e.g.[8,16,17]. It starts with a collection of DCs each of which is exposed to antigens (objects) and environmental signals. Below, we describe a novel approach for generating signals from the feature vectors. Then, we show how the DCA algorithm utilizes these signals to detect SMS spam messages.

2.2.1. Generation of DCA Signals

In DCA algorithm, there are three types of signals: $PAMP$, $Danger$ and $Safe$. The $PAMP$ signal is a measure of confidence that the antigen represents a spam. The $Danger$ signal is a measure which indicates a potential abnormality. Its value increases as the confidence of the monitored system being in abnormal status increases accordingly. Finally, the $Safe$ signal is a measure that increases in value in conjunction with legitimate messages. It represents a confidence indicator of normal, predictable or steady-state system behavior. To generate these required signals, we combined the outputs of two different machine learning algorithms: Naïve Bays (NB) and Support Vector Machine (SVM). The pseudo-code of this process for signal generation is outlined in Algorithm 1. For a particular message, each classifier takes the feature vector representing the message as input and generates a decision with a confidence level. Since the $PAMP$ signal indicates high level of assurance of an anomalous situation, it is generated using the highest confidence level of the two classifiers when both agree that the antigen is spam. The second type of signals is the presence of $Danger$ signals, which may or may not indicate an anomalous situation. However, the probability of an anomaly is higher than under normal circumstances. Hence, we used the average confidence level of the two classifiers when both disagree on the antigen classification. Finally, the presence of the $Safe$ signal indicates that no anomalies are present. In our case, if the two classifiers agreed that the antigen is non-spam, we utilized the highest confidence level of the two classifiers to be the $Safe$ signal. The derived signals and associated antigens passed to the DCA algorithm as input.

2.2.2. Dendritic Cell Algorithm

A high level view of the main steps in the DCA algorithm is shown in Algorithm 2. This algorithm starts with a population of dentritic cells (DCs)[8]. Each DC has a different lifespan which is initialized to some random value then changes over time based on the exposure to antigens and signals. The combination of signals and antigen temporal correlation and diversity of the DC population is responsible for the detection capability of the DCA. The maximum number of antigens that should be collected by a single DC is determined by concentration of co-stimulatory molecules (CSM) which is initially assigned randomly to each DC. When a threshold value of the CSM is reached, the DC is

migrated and transformed to a mature or semi-mature state. The transformation is based on the overall abnormality of signals seen by a dendritic cell which is denoted as K. At a particular exposure n, the impact of the three types of signals on CSM and K is calculated using the following formulas:

$$\Delta CSM = PAMP_n \times wc_p + Danger_n \times wc_d + Safe_n \times wc_s \tag{1}$$

$$\Delta K = PAMP_n \times wk_p + Danger_n \times wk_d + Safe_n \times wk_s \tag{2}$$

where $PAMP_n$, $Danger_n$, and $Safe_n$ are the input signals, wc_p, wc_d, and wc_s are weights associated with CSM, and wk_p, wk_d, and wk_s are weights associated with K. DCs are classified as mature or semi-mature based on the accumulated values of CSM and K as shown in Figure 1.

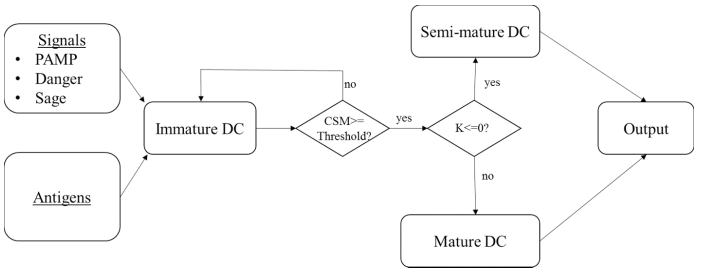

Fig. 1. Classification of DC as mature or semi-mature

The final decision to classify an antigen as *Spam* or *Legitimate* is made based on the number of DCs that are fully mature. This is done by computing a mature context antigen value ($MCAV$). This value gives a probability of a pattern being anomalous. The closer this value to 1, the greater the probability that the antigen is anomalous. To overcome the problem of antigen deficiency and to ensure that it appears in several contexts, each antigen is sampled multiple times using the antigen multiplier parameter of the DCA[17]. The DCA calculates the MCAV value for each antigen type using the following formula:

$$MCAV = \frac{M_i}{\sum Ag} \tag{3}$$

where i refers to the antigen type (spam), M_i refers to the number of times that antigen appears in the mature context and $\sum Ag$ is the total number of antigens. The $MCAV$ value is then used to classify the SMS by comparing it to an anomaly threshold that is calculated from:

$$at = \frac{an}{tn} \tag{4}$$

where at is the derived anomaly threshold, an is the number of anomalous data items and tn is the total number of data items. The classification rule applied on the i-th message is as follows:

$$f(x) = \begin{cases} \text{Spam}, & if MCAV > at \\ \text{Legitimate}, & otherwise \end{cases}$$

3. Experimental Work

3.1. SMS Datasets

We used two datasets to evaluate and compare the effectiveness of the proposed short message detection model. These datasets are publicly available and widely used in some other published work in the literature. Table 3 shows a brief summary of these datasets and detailed descriptions are presented next.

Table 3. Benchmark spam filtering datasets (total number of SMS instances, number of spam instances, number of legitimate instances, number of tokens per messages (TPM)).

Dataset#	Description	# SMS instances	# Spam instances	# Legitimate instances	# TPM
Dataset1	SMS Spam Corpus V.0.1 Big	1,324	322	1,002	15.72
Dataset2	SMS Spam Collection V.1	5,574	747	4,827	14.56

3.1.1. Dataset#1: SMS Spam Corpus V.0.1 Big

This corpus is a collection of 1,002 legitimate messages and 322 spam SMSs in English language. The legitimate SMS messages were randomly selected from the National University of Singapore (NUS) SMS corpus (10,000 legitimate SMSs) and the Jon Stevenson corpus (202 legitimate SMSs). The spam messages were collected manually from the Grumbletext Website, which is a public UK forum where users claims SMS spam messages. The average word length is 4.44 characters and the average number of words per message is 15.72[5]. This dataset is available at (`http://www.esp.uem.es/jmgomez/smsspamcorpus/`) and has been used in[5,7,18].

3.1.2. Dataset#2: SMS Spam Collection V.1

This corpus is a collection of spam and legitimate messages publicly available in raw format at (`http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/`) and is also hosted at the UCI machine learning repository. There are a total of 5,574 SMS messages in English gathered from four free or free for research sources: Grumbletext Website (425 SMS), Caroline Tag's PhD Theses (450 SMS), National University of Singapore (3,375 SMS) and Jon Stevenson Corpus (1,324 SMS). The corpus has a total of 4,827 legitimate messages and 747 spam messages. This corpus is described and analyzed in[4] and has been recently used in[19].

3.2. Evaluation Measures

The effectiveness is evaluated in terms of the percentage detection accuracy which is calculated from:

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \times 100 \tag{5}$$

where ACC is the accuracy, TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives. We also computed the percentage recall (REC), precision (PRE) and F-measure (F) for each category. Moreover, we computed the area under the ROC curve (AUC).

3.3. Experiments and Discussions

We first performed a series of experiments to evaluate the individual feature sets extracted from both datasets. Two types of machine learning algorithms are used: Support Vector Machine (SVM) and Naïve Bayes (NB). The results are shown in Table 4 and 5 for SVM and NB, respectively. The performance is recorded for 10-fold cross validation in terms of the precision (PRE), recall (REC) and F-measure (F) for each message category. The tables also show the percentage overall accuracy (ACC) and the area under the ROC curve (AUC) for each case. Analyzing these results, we found that there are two dominating feature sets with very high AUCs. These feature sets are the 'Spam Words (SW)' and 'Metadata (MD)'. They are more relevant to the classification process and their combination may yield better results. We then merged the two feature sets and rebuilt the classifiers to find out that this combination resulted in improving the effectiveness of both classifiers on both datasets. From the computational prospective, it will be better to combine only two feature sets rather than combining all the feature sets.

In order to demonstrate the effectiveness of the proposed DCA-based algorithm for SMS spam detection, we carried out the experiment again for both datasets. To adjust the DCA parameters, we ran several experiments with different values for the number of DCs, the Antigen Multiplier, and the signal weights. To manage the paper space, we only provide the best performance attained in Table 6 and the corresponding parameters are listed in Table 7. For the sake of comparison, we also show the best results obtained for SVM and NB in Table 6. It can observed that significant improvement is achieved by applying the proposed approach yet with only two most relevant feature sets.

Table 4. SVM classification results

		Spam			Legitimate				
Dataset	Feature Set	*PRE*	*REC*	*F*	*PRE*	*REC*	*F*	*AUC*	*ACC*
	URL	0.933	0.138	0.235	0.788	0.998	0.881	0.567	78.85
	Spam words (SW)	0.985	0.810	0.887	0.945	0.996	0.970	0.983	95.16
	Emotion symbols	0.000	0.000	0.000	0.763	1.000	0.865	0.500	75.68
	Special characters	0.000	0.000	0.000	0.762	0.998	0.864	0.606	75.53
Dataset1	All tags	0.689	0.503	0.576	0.857	0.926	0.890	0.717	82.33
	First and last terms POS	0.000	0.000	0.000	0.763	1.000	0.865	0.500	75.68
	Metadata (MD)	0.854	0.843	0.847	0.951	0.954	0.953	0.967	92.60
	Function words	0.579	0.497	0.530	0.851	0.887	0.868	0.845	79.23
	Combined(SW,MD)	0.978	0.871	**0.921**	0.962	0.994	**0.977**	**0.993**	**96.45**
	URL	0.956	0.144	0.248	0.883	0.999	0.937	0.571	88.43
	Spam words (SW)	0.922	0.757	0.831	0.964	0.990	0.977	0.959	95.89
	Emotion symbols	0.000	0.000	0.000	0.866	1.000	0.928	0.500	86.60
	Special characters	0.000	0.000	0.000	0.866	1.000	0.928	0.399	86.60
Dataset2	All tags	0.587	0.162	0.254	0.883	0.982	0.930	0.541	87.17
	First and last terms POS	0.000	0.000	0.000	0.866	1.000	0.928	0.500	86.60
	Metadata (MD)	0.712	0.456	0.554	0.920	0.972	0.945	0.887	90.22
	Function words	0.000	0.000	0.000	0.866	1.000	0.928	0.487	86.60
	Combined(SW,MD)	0.914	0.775	**0.838**	0.966	0.989	**0.977**	**0.973**	**96.02**

Table 5. Naïve Bayes classification results

		Spam			Legitimate				
Dataset	Feature Set	*PRE*	*REC*	*F*	*PRE*	*REC*	*F*	*AUC*	*ACC*
	URL	0.961	0.140	0.240	0.789	0.998	0.881	0.567	78.85
	Spam words (SW)	0.935	0.923	**0.928**	0.976	0.979	**0.978**	**0.983**	**96.60**
	Emotion symbols	0.240	1.000	0.387	0.600	0.013	0.025	0.500	25.30
	Special characters	0.525	0.221	0.305	0.795	0.937	0.860	0.753	76.36
Dataset1	All tags	0.553	0.610	0.556	0.847	0.788	0.787	0.731	74.31
	First and last terms POS	0.615	0.419	0.497	0.836	0.920	0.876	0.801	79.83
	Metadata (MD)	0.653	0.894	0.752	0.963	0.847	0.901	0.948	85.88
	Function words	0.571	0.545	0.556	0.860	0.870	0.865	0.848	79.15
	Combined(SW,MD)	0.855	0.949	0.899	0.984	0.949	0.966	**0.983**	94.79
	URL	0.948	0.143	0.248	0.883	0.999	0.937	0.500	88.43
	Spam words (SW)	0.737	0.863	0.794	0.978	0.952	0.965	0.960	94.03
	Emotion symbols	0.141	1.000	0.247	1.000	0.059	0.111	0.529	18.50
	Special characters	0.080	0.012	0.021	0.866	0.985	0.922	0.731	85.47
Dataset2	All tags	0.446	0.498	0.470	0.921	0.904	0.912	0.712	84.95
	First and last terms POS	0.692	0.169	0.270	0.885	0.988	0.934	0.767	87.82
	Metadata (MD)	0.548	0.809	0.653	0.968	0.896	0.931	0.925	88.45
	Function words	0.000	0.000	0.000	0.866	1.000	0.928	0.822	88.60
	Combined(SW,MD)	0.835	0.863	**0.848**	0.979	0.973	**0.976**	**0.967**	**95.86**

Table 6. Comparison of DCA with best performance of SVM and NB

		Spam			Legitimate				
Dataset	Approach	*PRE*	*REC*	*F*	*PRE*	*REC*	*F*	*AUC*	*ACC*
	Proposed	**1.000**	**0.991**	**0.995**	**0.997**	**1.000**	**0.999**	**0.999**	**99.77**
Dataset1	SVM	0.978	0.871	0.921	0.962	0.994	0.977	0.993	96.45
	NB	0.855	0.949	0.899	0.984	0.949	0.966	0.983	94.79
	Proposed	**1.000**	**0.996**	**0.998**	**0.999**	**1.000**	**1.000**	**0.999**	**99.95**
Dataset2	SVM	0.914	0.775	0.838	0.966	0.989	0.977	0.973	96.02
	NB	0.835	0.863	0.848	0.979	0.973	0.976	0.967	95.86

4. Conclusions

With the evolution of mobile technology and the increased dependence on smart devices, the number of spam SMS messages is unprecedentedly growing. Spam is not only annoying but it can be a vehicle for more severe security threats and information leakage as well. To control this problem, we analyzed and evaluated several feature

Table 7. DCA best parameters used.

Parameters	Values for Dataset1	Values for Dataset2
Number of DCs	40	30
Antigen multiplier	80	50
Signals weights	$\begin{cases}\Delta CSM = 2 \times PAMP + Safe + Danger \\ \Delta K = 2 \times PAMP - 3 \times Safe + Danger\end{cases}$	

sets, which can be easily extracted from the received messages, using two machine learning algorithms. We also explored the impact of combing the two most relevant sets on the performance of the machine learning algorithms. Subsequently, we developed a novel approach based on DCA that fuses the output from two classifiers. The empirical results showed significant improvement can be achieved when applying the proposed approach (with close to 100% accuracy). As future work, we are planning to compare it with other models and test it on different datasets.

Acknowledgment

The second author would like also to acknowledge the support provided by King Abdulaziz City for Science and Technology (KACST) through the Science & Technology Unit at King Fahd University of Petroleum & Minerals (KFUPM) for funding this work under project No. 11-INF1658-04 as part of the National Science, Technology and Innovation Plan.

References

1. Sanou, B. The world in 2014: ICT facts and figures. 2014. `https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf`.
2. Baldwin, C. 350,000 different types of spam SMS messages were targeted at mobile users in 2012. 2013. `http://www.computerweekly.com/news/2240178681/`.
3. Sohn, D.N., Lee, J.T., Han, K.S., Rim, H.C. Content-based mobile spam classification using stylistically motivated features. *Pattern Recognition Letters* 2012;**33**(3):364–369.
4. Almeida, T.A., Hidalgo, J.M.G., Yamakami, A. Contributions to the study of SMS spam filtering: new collection and results. In: *Proceedings of the 11th ACM Symposium on Document Engineering*. 2011, p. 259–262.
5. Cormack, G.V., Gómez Hidalgo, J.M., Sánz, E.P. Spam filtering for short messages. In: *Proceedings of the 16th ACM Conference on Information and Knowledge Management*. 2007, p. 313–320.
6. Delany, S.J., Buckley, M., Greene, D. SMS spam filtering: methods and data. *Expert Systems with Applications* 2012;**39**(10):9899–9908.
7. Gómez Hidalgo, J.M., Bringas, G.C., Sánz, E.P., García, F.C. Content based SMS spam filtering. In: *Proceedings of the ACM Symposium on Document Engineering*. 2006, p. 107–114.
8. Greensmith, J., Aickelin, U., Cayzer, S. Detecting danger: The dendritic cell algorithm. In: *Robust Intelligent Systems*. 2008, p. 89–112.
9. Wright, W.R., Chin, D.N. Personality profiling from text: Introducing part-of-speech n-grams. In: *User Modeling, Adaptation, and Personalization*. Springer; 2014, p. 243–253.
10. Kim, M.H., Compton, P. Improving the performance of a named entity recognition system with knowledge acquisition. In: *Knowledge Engineering and Knowledge Management*. Springer; 2012, p. 97–113.
11. Hull, D.A. Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science - Special Issue: Evaluation of Information Retrieval Systems* 1996;**47**(1):70–84.
12. Koppel, M., Ordan, N. Translationese and its dialects. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics; 2011, p. 1318–1326.
13. Pennebaker, J.W., Francis, M.E., Booth, R.J. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates* 2001;**71**:2001.
14. Argamon, S., Levitan, S. Measuring the usefulness of function words for authorship attribution. In: *Proc. ACH/ALLC Conference*. 2005.
15. Gilner, L., Morales, F. Function words. 2014. `http://www.sequencepublishing.com`.
16. Greensmith, J., Aickelin, U. The deterministic dendritic cell algorithm. In: *Artificial Immune Systems*. Springer; 2008, p. 291–302.
17. Huang, R., Tawfik, H., Nagar, A. Artificial dendritic cells algorithm for online break-in fraud detection. In: *Proceedings of the 2nd IEEE International Conference on Developments in eSystems Engineering (DESE)*. 2009, p. 181–189.
18. Cormack, G.V., Hidalgo, J.M.G., Sánz, E.P. Feature engineering for mobile (SMS) spam filtering. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*; 2007, p. 871–872.
19. Almeida, T., Hidalgo, J.M.G., Silva, T.P. Towards sms spam filtering: Results under a new dataset. *International Journal of Information Security Science* 2013;**2**(1):1–18.